Internet of Things: A survey related to various recent Architectures and Platforms available

Mr.Sharwari Satish Solapure
Assistant Professor, CSE Department
Walchand College of Engineering
Sangli-India
solpaure_ss@yahoo.co.in

Dr.Harish Kenchannavar
Associate Professor in CSE Department,
Gogate Institute of Technology,
Belgavi-India
harishhk@git.edu

Abstract—IP-based Internet is the largest network in the world therefore; there are excessive steps towards connecting Wireless Sensor Networks (WSNs) to the Internet. It is popularly known as to IoT (Internet of Things). IETF has developed a suite of protocols and open standards for accessing applications and services for wireless resource constrained networks such as IoT. Development of application requires standardized architecture and platform for design and analysis of new ideas. This paper provides a brief awareness about recent IoT architectures and platforms. It is also discussed some of the gaps issues of the platforms related to usability of the user. This helps researcher to select a particular platform according to need.

***Index Terms*— Internet of Things (IoT), 6LoWPAN CoAP, MQTT, RPL, SOA, SDWN.REST API**

I.INTRODUCTION:

"Internet of Things "(IoT) is a widely used term, but it is difficult to understand as it is composed by number of technologies. In the literature, several definitions of IoT overlap with each other. IoT is not a one technology; it is a concept where number of different things connected to each other such as embedded sensors, and camera nodes, near field communication nodes etc. The IoT, uses dissimilar methods for communication which need to be revised in order to address the matters of IoT applications such as energy efficiency, security, and reliability. IoT manages connectivity techniques for the needs of the IoT applications. It is mentioned in [1], wired and wireless technologies: Ethernet,Wi-Fi, Personal Area Network, IoT6 , ZigBee has adopted a protocol stack from IETF. The stack consist IEEE802.15.4 PHY-MAC, 6LoWPAN, RPL, and CoAP, and it fulfils the requirements of constrained devices.The important characteristics of the IoT are as follows:

• Interconnectivity: Any device can be connected with the worldwide information and communication infrastructure.

•Things-related services: Capability of providing things-related services for constraint devices/things.

• Heterogeneity: IoT devices are based on various hardware platforms and different networks.

•Dynamic changes: The device changes their states dynamically, e.g., sleeping and waking up, connected or disconnected also context: location and speed.

• Massive scale: The number of devices handled for communication with respect to total devices connected to the Internet.

Several hardware and software architectures of IoT are discussed in many research papers. Here In this paper some of the latest concepts related to architecture and platform of IoT are discussed in brief. It helps for developer for latest application development. Paper layout is as follows: Section I- Architectures, Section II- Software platform, Section III – Gap analysis in between Platforms and at last Conclusion.

II.ARCHITECTURES:

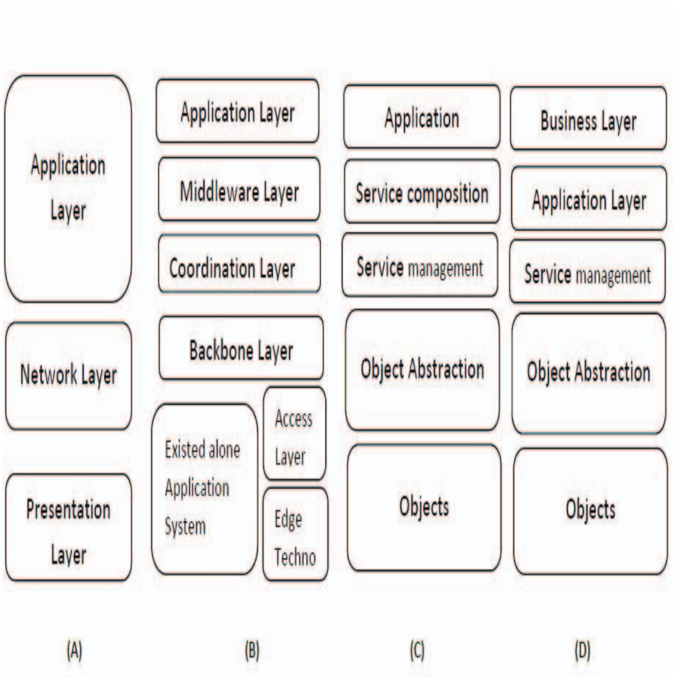
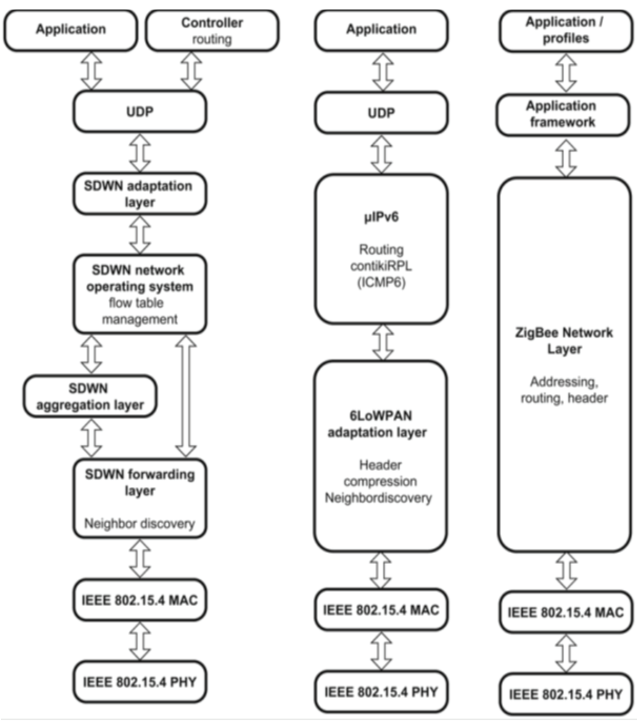


Fig. 2.1- IoT architecture. (A) Three-layer. (B) Middle-ware used. (C) Service of Architecture used. (D) Simple Five-layer

The IoT should be able to connect billions or trillions of heterogeneous devices over the Internet, so there is a serious need for a flexible or easily moldable layered architecture. However numbers of existed architectures are not up to the mark. There are several schemes who try to design a collective architecture based on the analysis of the needs of scientists and the business. The basic model is 3-layer architecture is presented in [2] which is shown in Fig. 2.1 (A) consisting Application layer, Network Layer and Perception Layer.
In the latest literature, several other models have been proposed [2] which adds other aspects to the IoT architecture as shown in Fig. 2.1 (B, C, D) consisting 5-layer.
In paper [3] different solutions, centralized software defined wireless networking (SDWN), ZigBee and IPv6 over low-power wireless personal area networks (6LoWPAN) are compared and tested. It is shown in Fig.2.2. Paper result states that SDWN is more suitable for fixed position nodes and low mobility scenario. However, for dynamic situation a distributed solution like ZigBee and 6LoWPAN could work better.



(A) (B) (C)

Fig. 2.2- SDWN (A), IPv6 (B) and ZigBee (C)

Several IoT standards are present in [4] to support and shorten application programmers and service provider's jobs. Various types of groups have been formed to offer protocols in support of the IoT such as the World Wide Web Consortium (W3C), Internet Engineering Task Force (IETF), Institute of Electrical and Electronics Engineers (IEEE) and European telecommunications Standards Institute (ETSI). Fig.2.3 gives us a brief idea about most prominent protocols defined by these groups.
IoT protocols are classified in Fig. 2.3 into four general categories: application protocols, service discovery protocols, infrastructure protocols and other influential protocols. These are bundled together to deliver an IoT application. All standards may not be used by every IoT application.

Application Protocol		CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP RESET
Service Discovery		mDNS			DNS-SD		
Infrastructure Protocols	Routing Protocol	RPL					
	Network Layer	6LoWPAN			IPV4/IPV6		
	Link Layer	IEEE 802.15.4					
	Physical/ Device Layer	LTE-A	EPCglobal	IEEE 802.15.4	Z-WARE		
Influential Protocols		IEEE 188.3,IPSec			IEEE 1905.1		

Fig. 2.3- Standardization Efforts in support of IoT

The paper [4] mentions that all these protocols together are not having complete evaluation at present.
However, each of these protocols may perform well in explicit for particular use case or environment. It is unable to offer a distinct treatment for all IoT applications.
Table 2.4 delivers a comparison between all IoT application protocols with respect to other layer protocols and services.

Application Protocol	RESTful	Transport	Pub/Sub	Security
COAP	√	UDP	√	DTLS
MQTT	x	TCP	√	SSL
MQTT-SN	x	TCP	√	SSL
XMPP	x	TCP	√	SSL
AMQP	x	TCP	√	SSL
DDS	x	UDP	√	DTLS
HTTP	√	TCP	x	SSL

Table. 2.4- Comparison between IoT Protocols

Next step is to understand that how IoT building blocks used for functionality of the IoT. It is helpful for an application

development. It actually consist steps followed by an IoT application developer.
In Fig. 2.5 the elements used to deliver the functionality of the IoT are shown.

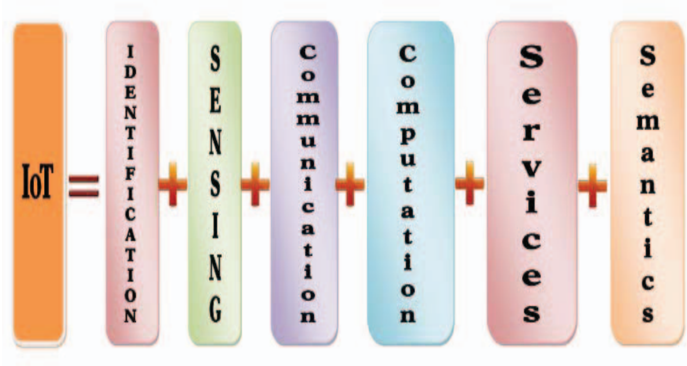


Fig. 2.5- The IoT Elements

Fig. 2.6-shows the classes of these elements and examples of each class.

BUILDING BLOCKS AND TECHNOLOGIES OF IOT

Iot Elements		Sample
Identification	Naming	EPC, uCode
	Addressing	IPv4,IPv6
Sensing		Smart Sensors, Wearable
		Sensing devices, Embedded
		Sensors, Actuators, RFID tag
Communication		RFID, NFC, UWB,
		Bluetooth, BLE, IEEE,
		802.15.4,Z-Wave,WiFi,
		WiFiDirect, LTE-A
Computation	Hardware	Smart Things, Arduino Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, Beagle Bone, Cubie Board, Smart Phones
	Software	OS (Contiki, TinyOS, LiteOS, RiotOS, Android) Cloud(Nimbits, Hadoop,Etc)
Services		Identity related(shipping),
		Information aggregation
		(Smart Grid), Collaborative-
		Aware(Smart Home),
		Ubiquitous(Smart City)
Semantics		RDF, OWL, EXI

Fig. 2.6- Building Blocks and Technologies of the IoT

III. Platforms:

Actually along with Hardware architecture, the IoT requires software architectures [4] that are able to use large amounts of information, queries, and computation. It can be possible by data/stream processing, filtering, aggregation and data mining. These all are sustained by communication standards such as HyperText Transfer Protocol (HTTP) and Internet Protocol (IP).

ELIoT, a development platform for Internet-connected smart devices is proposed in [5]. Unlike most solutions for the emerging "Internet of Things"(IoT), ELIoT allows programmers to implement functionality running *within* the networks of smart devices without necessarily leveraging the external Internet. ELIoT resolves the demand for efficient performance, *e.g.*reduced latency. To this end, ELIoT's programming model provides IoT-specific inter-process communication facilities. Its virtual machine-based execution satisfies the need of software reconfiguration and the heterogeneity.
Considering available Hardware and Software Architecture the development in the IoT application is thought-provoking because it includes handling of number of issues related to scalability and heterogeneity. The tasks have to be executed on the precise hardware along with its distribution to underlying network. Apart from this, developers have to handle deviations in application requirements and used devices.
Existing methods solve only limited challenges of IoT. In this paper [6] an approach for addressing the above challenges is proposed. The main contributions of this paper are: (1) a methodology for development which splits IoT application into different concerns and provides a conceptual framework for an application, (2) a framework development used to support stakeholders.
The development of framework is supporting scalability, heterogeneity and related complexity. It includes code generation, task-mapping, and additional techniques for automation.

Platforms: Next section some of the platforms are discussed in brief.

The paper [8] reviewed all the following IoT platforms. Mostly all of them are cloud based and open source. Few of them differ with centralized or decentralized approach. By referring paper [8] classification is done and presented here in table format. The Table 3.1 shows the details

Table 3.1: Classification of number of IoT Platforms

Name	Open source	Cloud Based	Centralized	Tools used	Link
AirVantage	√	√	×	*m2m. eclipse*	https://airvantage.net
Arkessa	×	√	×	MOSAIC	http://www.arkessa.com
ARM mbed	×	×	×	REST API	https://mbed.org
Carriots	×	√	×	REST API	https://www.carriots.com
DeviceCloud	×	√	×	REST API	http://www.etherios.com/products/ devicecloud
Devicehub.net	×	√	×	REST API	http://www.devicehub.net
EveryAware	√	×	√	REST API	http://www.everyaware.eu
EveryWare Device Cloud	×	√	×	REST API	http://www.eurotech.com
EvryThng	×	×	√	REST API	http://www.evrythng.com
Exosite	√	√	×		http://exosite.com
Fosstrack	√	√	×		https://code.google.com/p/fosstrak
GroveStreams	√-small users	√	×	REST API	https://grovestreams.com
Hub of All Things	×	×	×		http://hubofallthings.wordpress.com
Ericsson IoT-Framework	×	×	×	REST API	https://github.com/ ricssonResearch
IFTTT	×	×	×		https://ifttt.com
Kahvihub	√	×	×		http://github.com/uh-cs-iotlab/ kahvihub
LinkSmart TM	√	×	×		http://www.hydramiddleware.eu
MyRobots	×	√	×	REST API	https://www.myrobots.com
Niagara *AX*	×	×	×	OPEN API	httsp://www.niagaraax.com
Nimbits	√	√	×		http://www.nimbits.com
NinjaBlock	√	×	×	REST API	http://ninjablocks.com
Node-RED	√	×	×		http://nodered.org
OpenIoT	√	×	√		http://openiot.eu
OpenMTC	√	√	×		http://www.open-mtc.org
OpenRemote	√	×	√		http://www.openremote.org
Open.Sen.se	×	×	×	Funnel	http://open.sen.se
RealTime.io	×	√	×		https://www.realtime.io
SensorCloud	×	√	×	REST API	http://www.sensorcloud.com
SkySpark	×	√	×	REST API	http://skyfoundry.com/skyspark
Swarm	×	√	×	GUI	http://buglabs.net/products/swarm
TempoDB	×	√	×	REST API	https://tempo-db.com

Name	Open source	Cloud Based	Centralized	Tools used	Link
TerraSwarm	√	√	×		http://www.terraswarm.org
Thing Broker	×	×	√	REST API	http://www.magic.ubc.ca/wiki/pmwiki.php/ThingBroker
ThingSpeak	√-GPLv3	×	×		https://www.thingspeak.com
ThingWorx	×	√	×	REST API	http://www.thingworx.com
Sense Tecnic WoTkit	×	√	×		http://sensetecnic.com
Xively	×	√	×	OPEN API REST API	https://xively.com

Systematic research available in [7] estimates the capabilities of mostly popular IoT platforms. It is providing clear understanding of the current IoT platform scene by studying six selected commercial IoT platforms, and by passing on an architectural model which illustrates the main building blocks of an IoT platform in a broad way.

Axeda platform : The application development is handled rather than providing connectivity only; although it offers an agent into a physical product for managing the connection to the platform using the proprietary Axeda wireless protocol.
Digi's platform: It contains communication modules, gateways and a software platform. These three levels are very useful development. XBee communication is based on the industry standard ZigBee.
Eclipse M2M group: It pulls existing Eclipse projects, and provides protocols like MQTT and OMA-DM. It is having heavyweight embedded framework.
Thingsquare : A web-based IDE called "Thingsquare Code" is used to form applications for Mist online. It includes the compilation and circulation process of embedded software to things.
Thingworx: its main focus is on data management. It reduces complexity for non-technical users by providing mashup technologies
Xively web service: It agrees provision, activation and managing of devices. It uses a unique identity for each device and rights to craft and receive data on the platform

IV.GAP ANALYSIS:

All these platforms, both proprietary and open-source, on the basis of their facility come across the expectations of different IoT users. The evaluation is focused on usability of these platforms for IoT users in [8]. The evaluation is carried out in [8] as a gap analysis for current IoT platforms. The parameters used for that are used: support towards -heterogeneous hardware, data management, application developers, ecosystems, as well as dedicated marketplaces for the IoT.

All together, these capabilities present the needs of emerging IoT ecosystem, the device vendors, developers, platforms providers, service providers and the end-users.

CONCLUSION:

The above discussion indicates that the strong knowledge of architecture and platform is essential for developers. The gap analysis concept is also very important for developers when there are more constraints for a particular application.
The selection of a particular architecture along with proprietary or open-source platform is must. This discussion helps a lot for this selection.
In future rigorous survey required to formulate some of the standard architectures and platforms which can be easily used by researchers/developers.

REFERENCES:

[1] Editors -Ovidiu Vermesan,Peter Friess "Internet of Things – From Research and Innovation to Market Deployment " River Publishers.(Book)

[2] A.Al-Fuqaha, Mohsen Guizani,Mehdi Mohammadi, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications"- IEEE Communication Surveys and Tutorials, Vol. 17, No 4, 2015.

[3] Chiara Buratti, Andrea Stajkic, G.Gardasevic, S. Milardo, M. D. Abrignani, ”Testing Protocols for the Internet of Things on the EuWIn Platform” IEEE IoT Journal, Vol.3, No.1, 2016.

[4] Maria Palattella, Nicola Accettura, Xavier Vilajosana, Thomas W. Luigi Grieco, Gennaro Boggia, and Mischa Dohler, “Standardized Protocol Stack for the Internet of (Important) Things”, IEEE Communication Surveys and Tutorials, Vol. 15, No. 3, 2013.

[5] A. Sivieri and L. Mottola."Building Internet of Things software with ELIoT" in Computer Communications, 2016, DOI:j.comcom.2016.02.004

[6] PankeshPatela,DamienCassouba,ABBCorporate Research,India InriaLille-NordEurope,France ”Enabling high-level application development for the Internet of Things“- Journal of System and Software,Vol.103,Elesvier-2015, https://dx.doi.org/10.1016/j.jss.2015.01.027

[7] Weinberger,Köhler,Marcus,Wörner,Dominic and Wortmann, “Platforms for the IoT-An Analysis of Existing Solutions“ in Felix 5th Bosch Conference on Systems and Software Engineering (BoCSE)-2014

[8] J.Mineraud,Oleksiy,Xiang Su,Sasu “A gap analysis of Internet-of-Things platforms”-Elesvier March 2016, https://dx.doi.org/10.1016/j.comcom.2016.03.015